


언니 그 필터 모예요? 사용 매뉴얼



야 너두 필터 쓸 수 있어!

박시현

양수영

이상은

INDEX

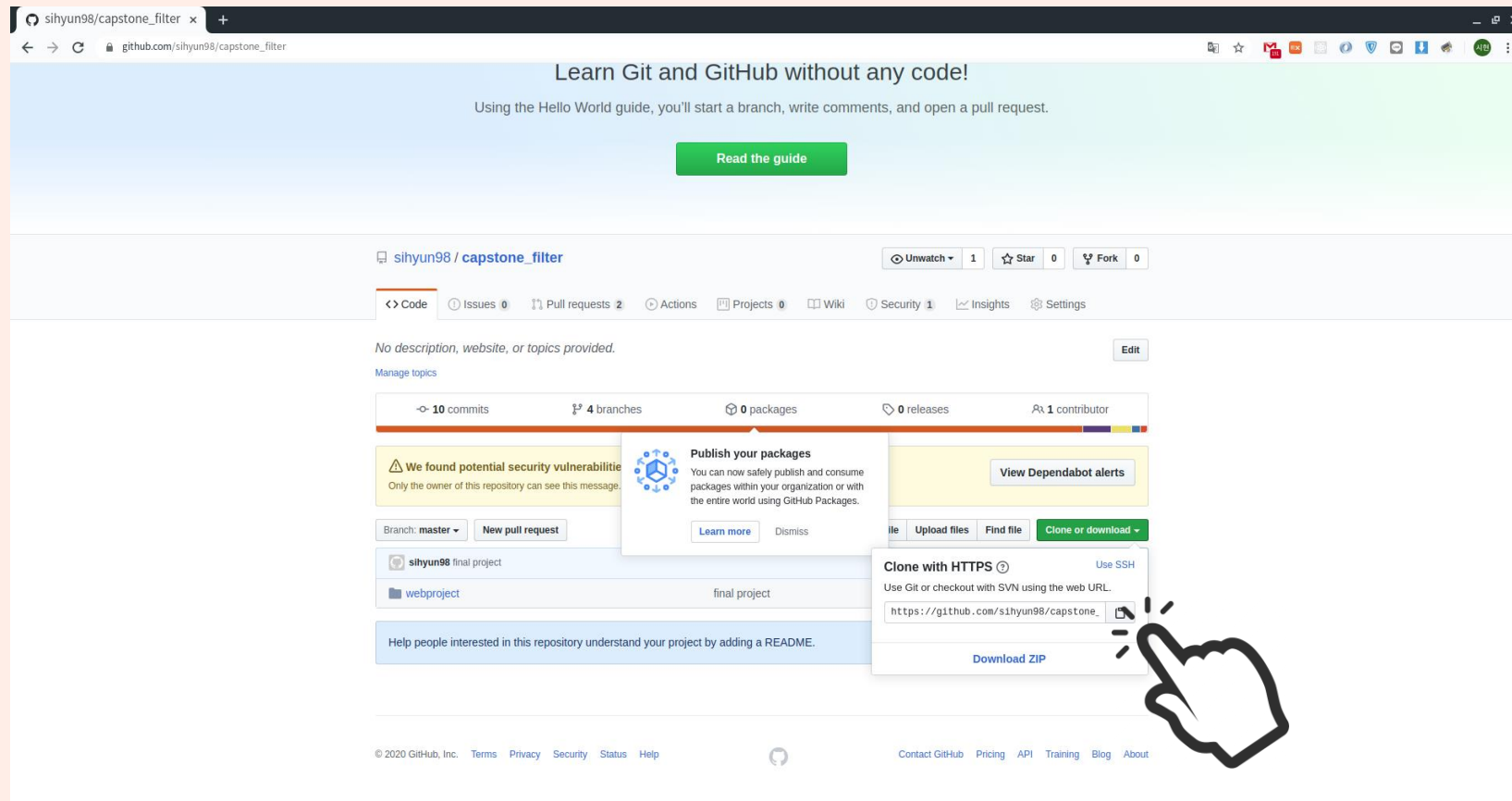
1. 서비스 설치 및 실행

2. 웹 서비스 사용

1. 서비스 설치 및 실행

프로젝트 깃허브 다운로드

https://github.com/sihyun98/capstone_filter



프로젝트 github에 들어간다.

Clone or download 버튼을
클릭한 후, HTTPS 또는 SSH
주소를 복사한다.

또는 zip 파일을 받아 압축
해제를 한다.

1. 서비스 설치 및 실행

서비스 설치

1. 복사한 주소를 로컬에 clone 받는다.

```
sihyun@sihyun-900X5N:~/바탕화면$ git clone https://github.com/sihyun98/capstone_filter.git
'capstone_filter'에 복제합니다...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 13987 (delta 0), reused 9 (delta 0), pack-reused 13969
오브젝트를 받는 중: 100% (13987/13987), 47.76 MiB | 12.04 MiB/s, 완료.
델타를 알아내는 중: 100% (5083/5083), 완료.
```

2. webproject/models/research 경로에 python 경로를 추가한다.

```
sihyun@sihyun-900X5N:~/바탕화면/capstone_filter/webproject$ cd models/research
sihyun@sihyun-900X5N:~/바탕화면/capstone_filter/webproject/models/research$ export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`:/slim
```

1. 서비스 설치 및 실행

로컬 경로 수정

3. webproject/object_detection_capstone/capstone_detection.py
에서 base_path를 로컬 경로로 변경한다.

```
33
34 base_path = '/home/sihyun/바탕화면/Demo/copy/object_detection_capstone/'
35 test_record_fname = base_path + 'data/annotations/test.record'
36 train_record_fname = base_path + 'data/annotations/train.record'
37 label_map_pbtxt_fname = base_path + 'data/annotations/label_map.pbtxt'
38 coco_label_map_fname = base_path + 'data/annotations/mscoco_label_map.pbtxt'
39
40 num_classes = get_num_classes(label_map_pbtxt_fname)
41 coco_num_classes = get_num_classes(coco_label_map_fname)
42
43 import os
44 import glob
45
46 # Path to frozen detection graph. This is the actual model that is used for the object detection.
47
48
49 pb_fname = base_path + 'frozen_inference_graph_12000_ssd_lite.pb'
50 pb_fname2 = base_path + 'frozen_inference_graph_coco.pb'
51
```

1. 서비스 설치 및 실행

서비스 설치

4. 가상환경을 위한 패키지를 다운받는다.

```
sihyun@sihyun-900X5N:~/바탕화면/capstone_filter$ pip3 install virtualenv
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/f0/55/f0de23cb61f2e90db3f8362f0e685a0bc04d0c4cb159b7a96cb7c4817d63/virtualenv-20.0.23-py2.py3-none-any.whl (4.7MB)
    100% |#####| 4.7MB 302kB/s
Collecting six<2,>=1.9.0 (from virtualenv)
  Cache entry deserialization failed, entry ignored
  Using cached https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Collecting importlib-resources>=1.0; python_version < "3.7" (from virtualenv)
  Downloading https://files.pythonhosted.org/packages/ce/06/94cb4028cbcb834c3a6a0c239ab37e76e293c75bed51a9cee2cb8337b47e/importlib_resources-2.0.1-py2.py3-none-any.whl
Collecting distlib<1,>=0.3.0 (from virtualenv)
Collecting appdirs<2,>=1.4.3 (from virtualenv)
  Using cached https://files.pythonhosted.org/packages/3b/00/2344469e2084fb287c2e0b57b72910309874c3245463acd6cf5e3db69324/appdirs-1.4.4-py2.py3-none-any.whl
Collecting importlib-metadata<2,>=0.12; python_version < "3.8" (from virtualenv)
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/98/13/a1d703ec396ade42c1d33df0e1cb691a28b7c08b336a5683912c87e04cd7/importlib_metadata-1.6.1-py2.py3-none-any.whl
Collecting filelock<4,>=3.0.0 (from virtualenv)
```

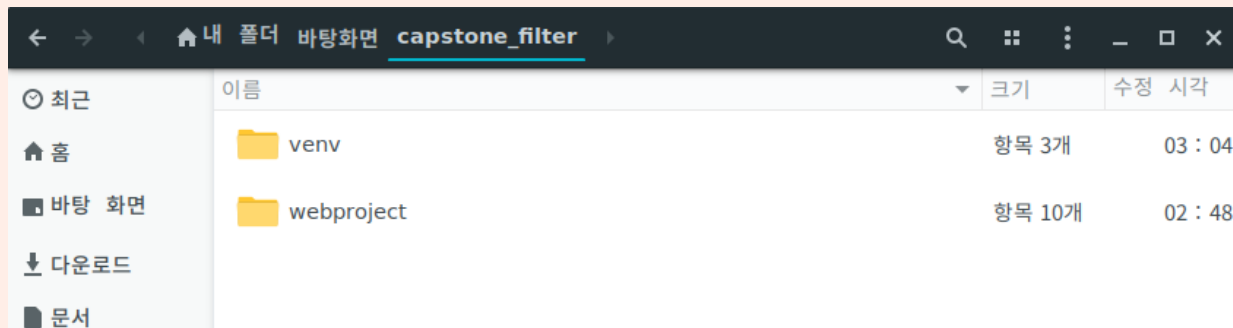
1. 서비스 설치 및 실행

서비스 설치

5. 가상환경을 생성한다.

```
sihyun@sihyun-900X5N:~/바탕화면/capstone_filter$ virtualenv venv
created virtual environment CPython3.6.9.final.0-64 in 230ms
  creator CPython3Posix(dest=/home/sihyun/바탕화면/capstone_filter/venv, clear=False, global=False)
  seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, via=copy, app_data_dir=/home/sihyun/.local/share/virtualenv/seed-app-data/v1.0.1)
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

6. 가상환경 폴더 생성을 확인한다.

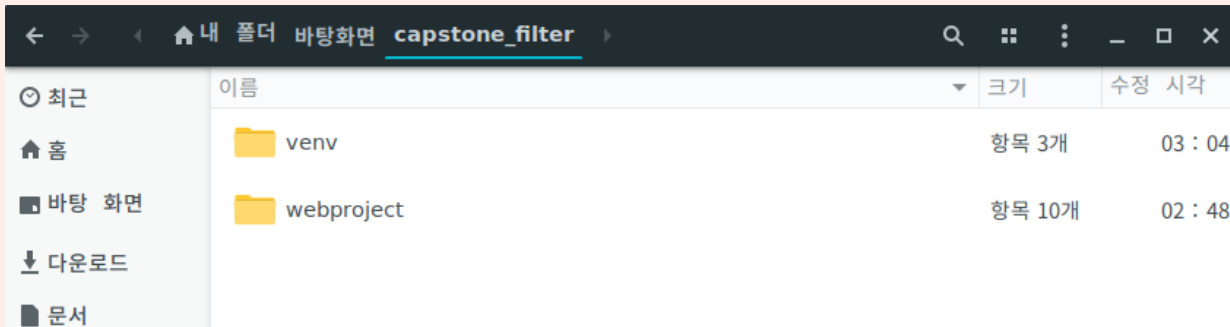


1. 서비스 설치 및 실행

서비스 실행

7. 가상환경 폴더에 requirements.txt의 패키지들을 설치한다.

8. 아래 directory에서 `source venv/bin/activate` 를 통해 가상환경을 실행한다.

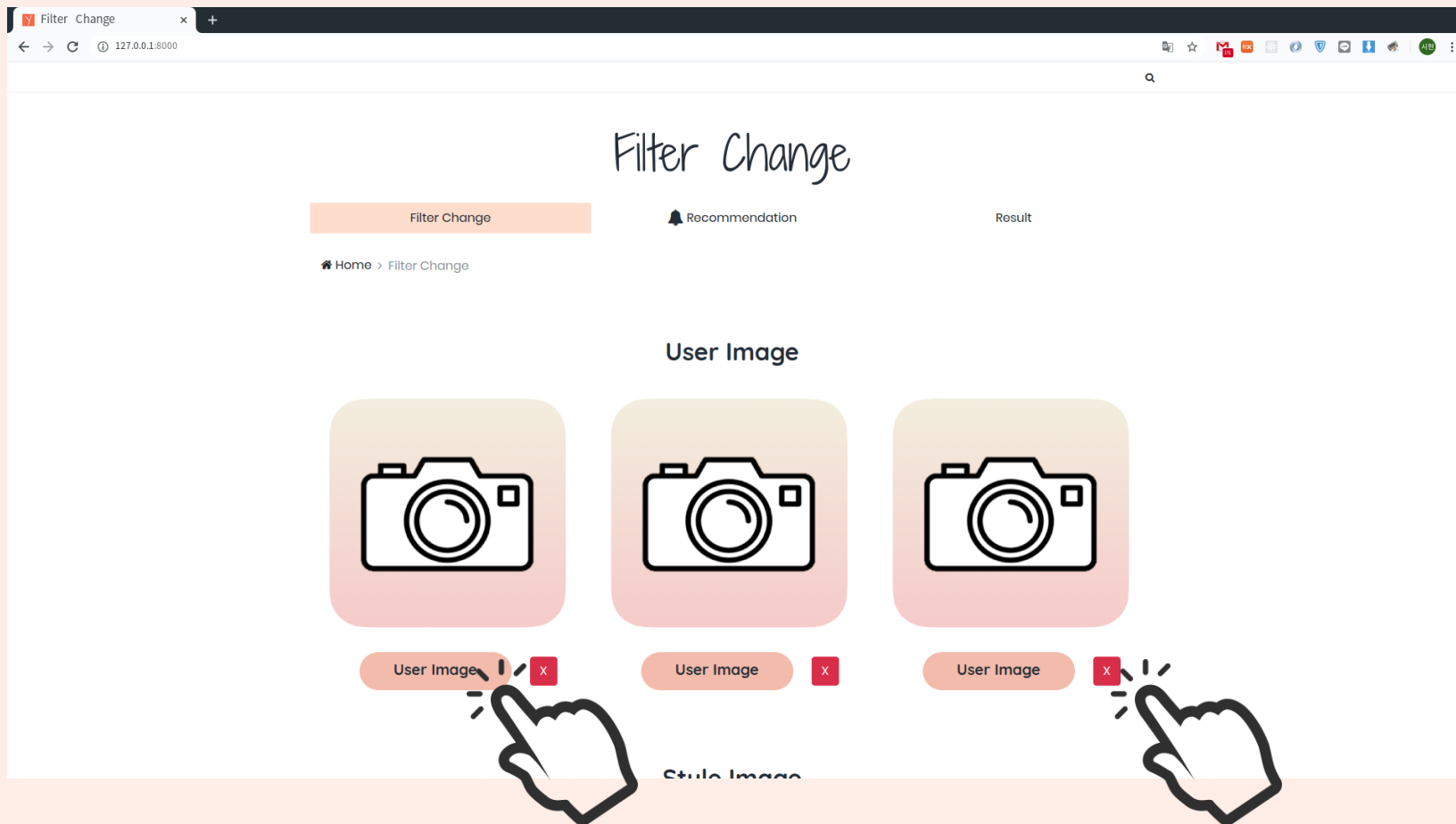


9. `cd webproject`로 경로 변경을 한다.

10. `python manage.py runserver`로 로컬 서버 실행을 한다.

2. 웹 서비스 사용

유저 이미지 및 스타일 이미지 업로드



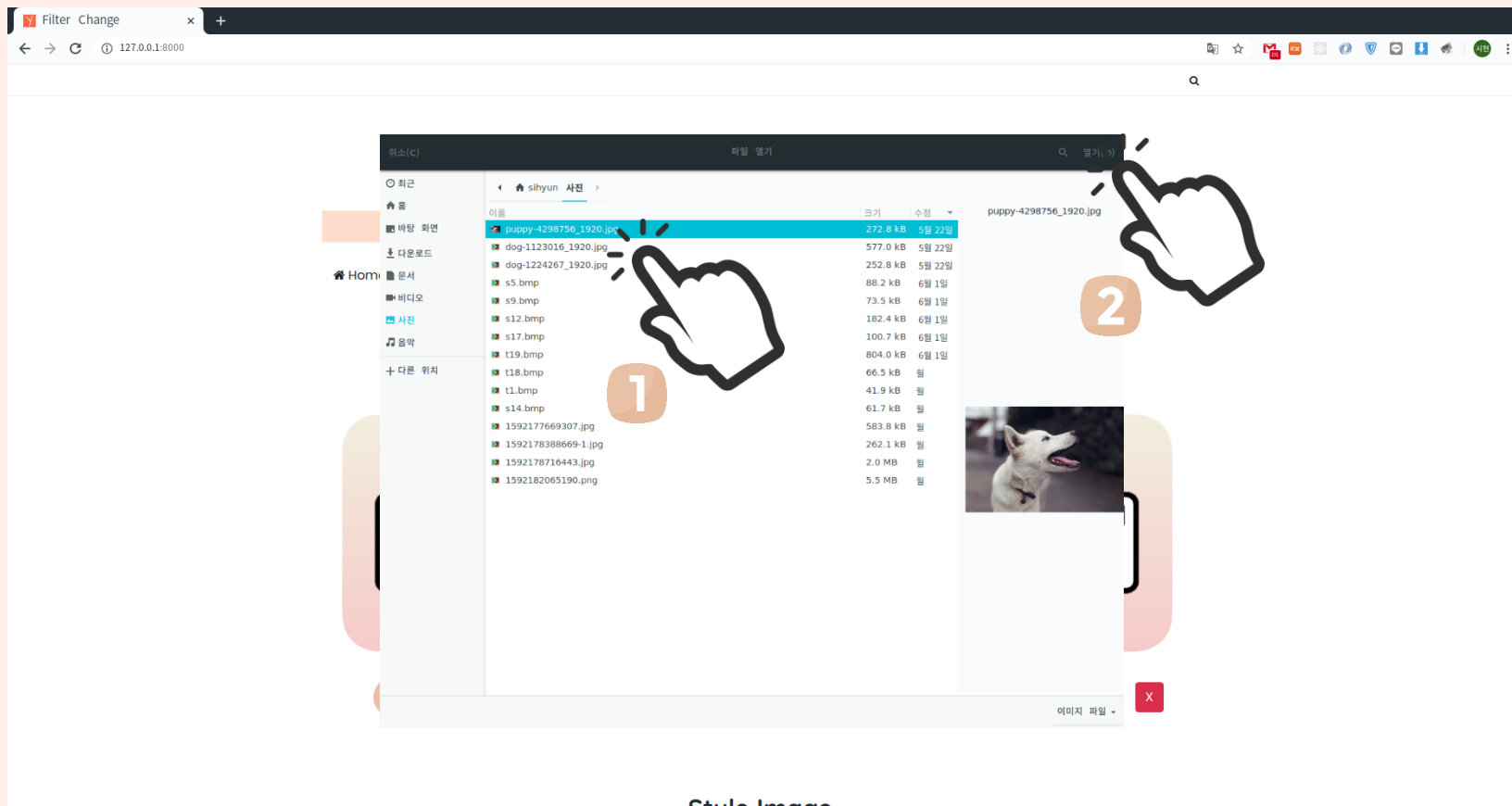
메인 화면 구성은 다음과 같다.

이미지를 업로드하기 위해서
User Image 버튼을 눌러
파일을 첨부한다.

필요 시, x버튼을 눌러
첨부한 이미지를 삭제할 수 있다.

2. 웹 서비스 사용

유저 이미지 및 스타일 이미지 업로드

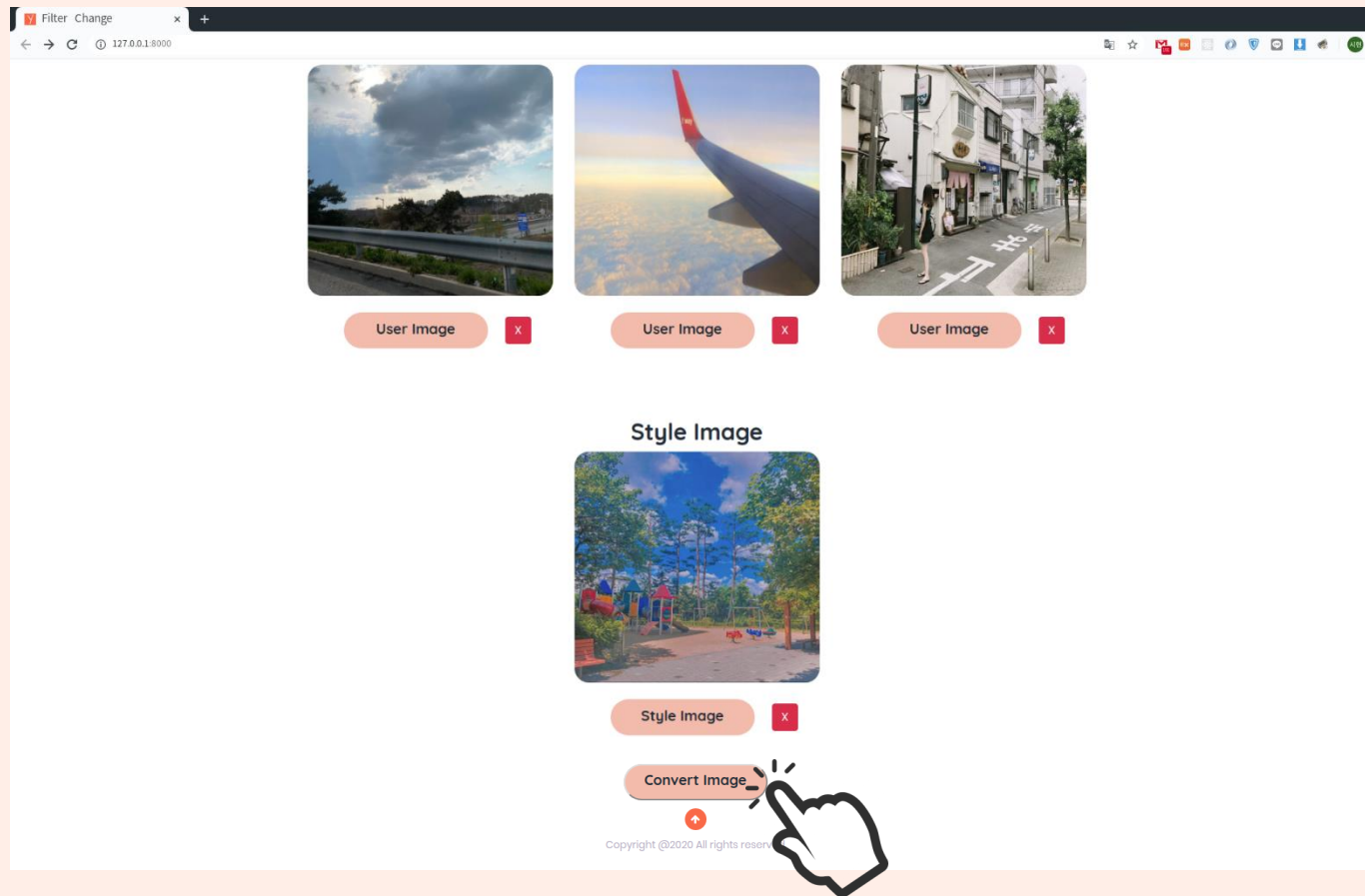


원하는 이미지를 선택한 후에
이미지를 열어
파일을 첨부한다.

이미지 형식의 파일만
업로드 가능한 것을
확인할 수 있다.

2. 웹 서비스 사용

유저 이미지 및 스타일 이미지 업로드

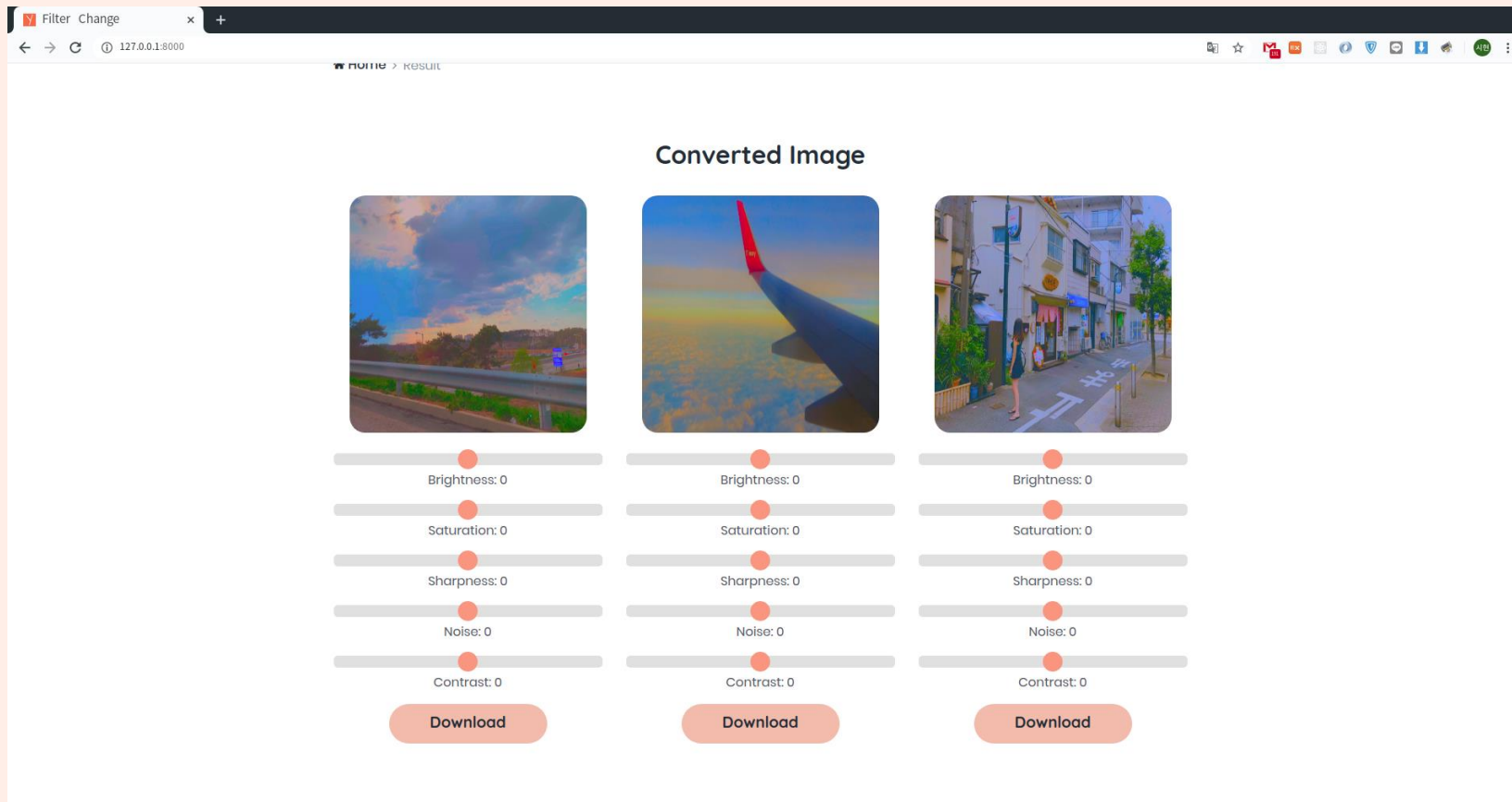


최대 3장의 유저 이미지 (User Image)와 1장의 스타일 이미지(Style Image)를 업로드 할 수 있다.

이미지 업로드 완료 후에는 Convert Image 버튼을 눌러 결과 이미지를 확인한다.

2. 웹 서비스 사용

이미지 변환

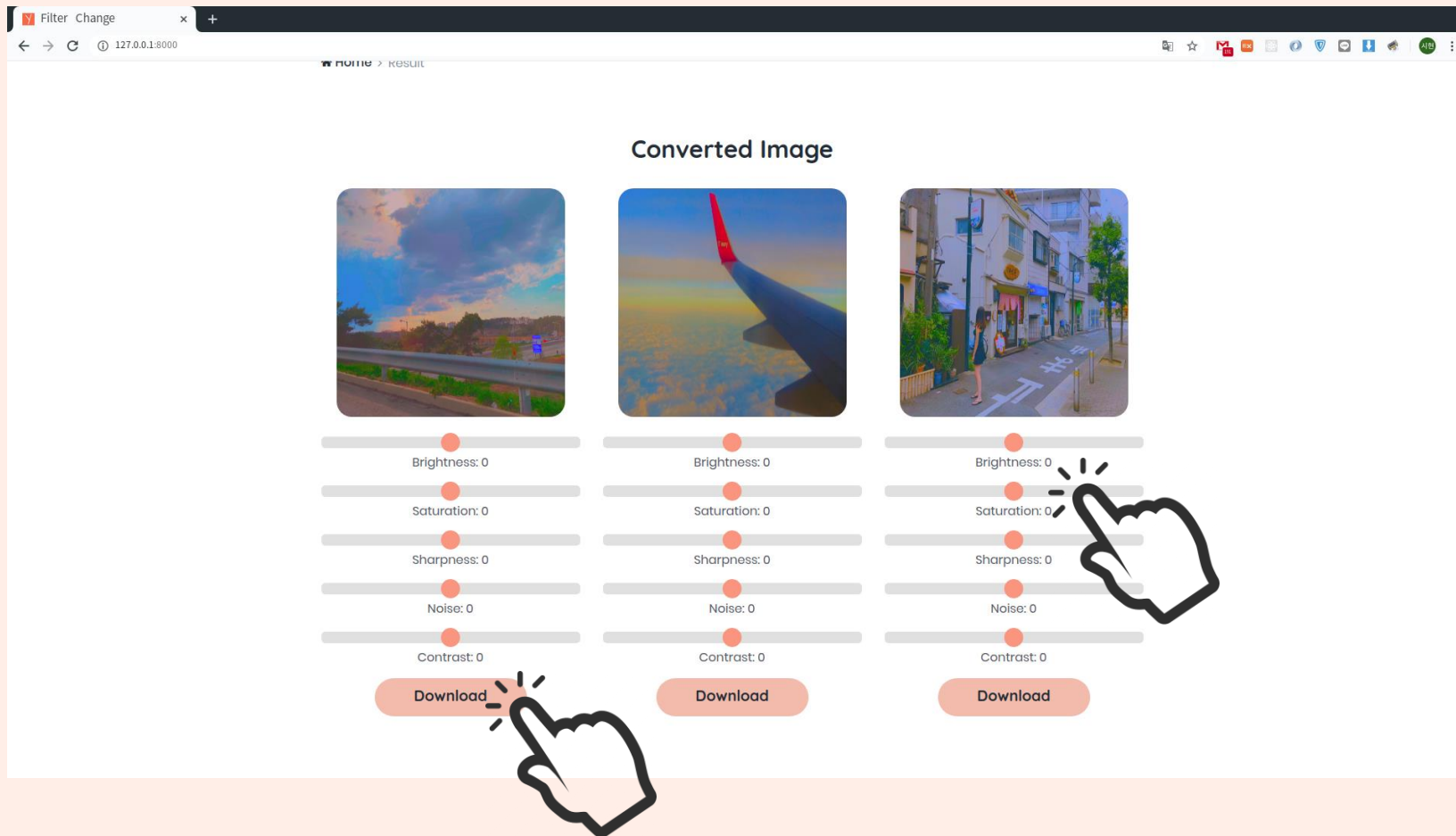


변환된 결과 이미지를
Result 탭에서 확인할 수 있다.

유저 이미지(User Image)
입력 개수에 따라 최대 3장의
결과 이미지를 얻을 수 있다.

2. 웹 서비스 사용

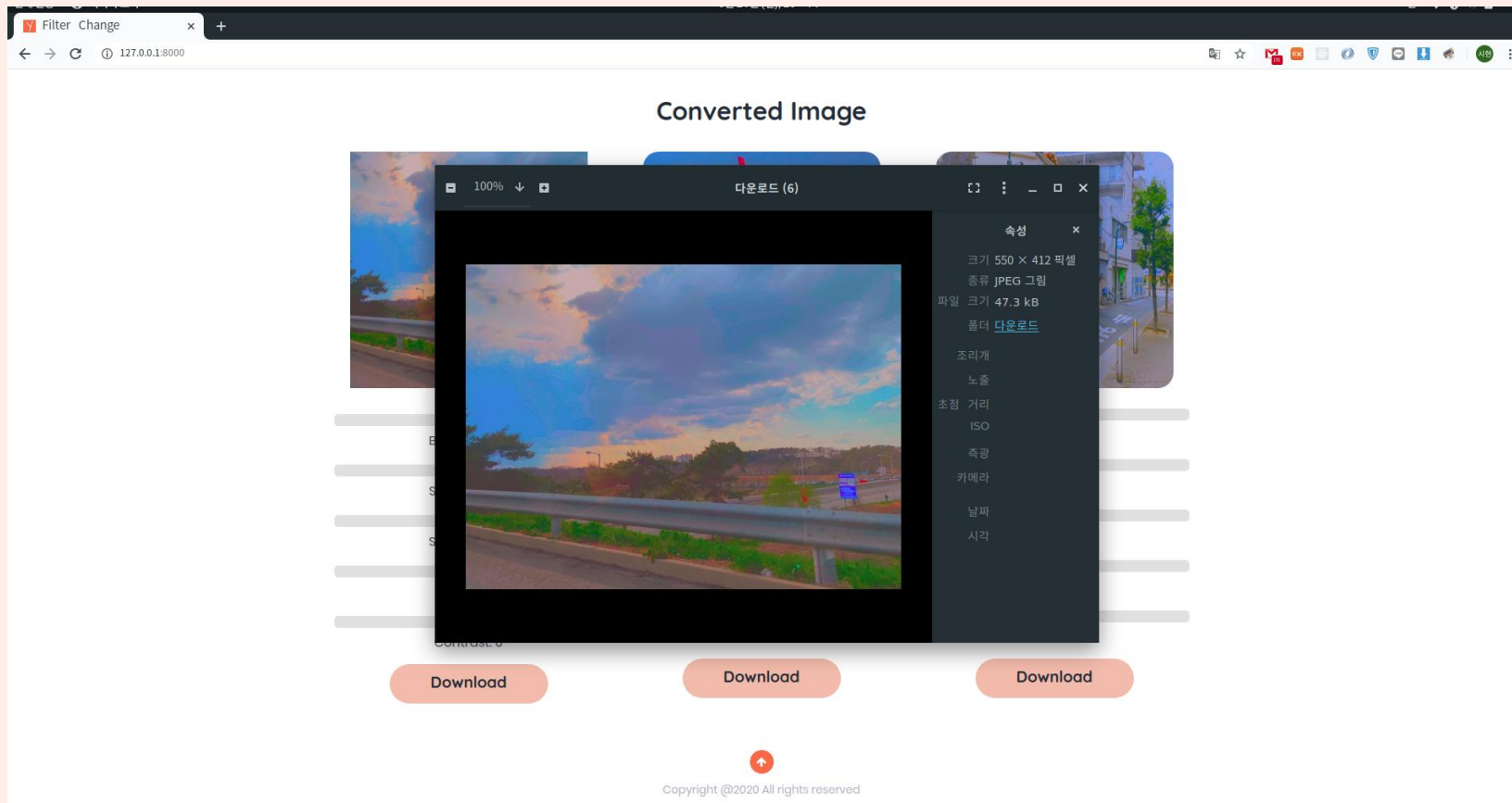
결과 이미지 다운로드



Result 탭에서 결과 이미지를 확인하고,
추가 보정을 완료한 후
다운로드(Download) 버튼을
눌러 해당 이미지를
다운로드할 수 있다.

2. 웹 서비스 사용

이미지 다운로드 확인

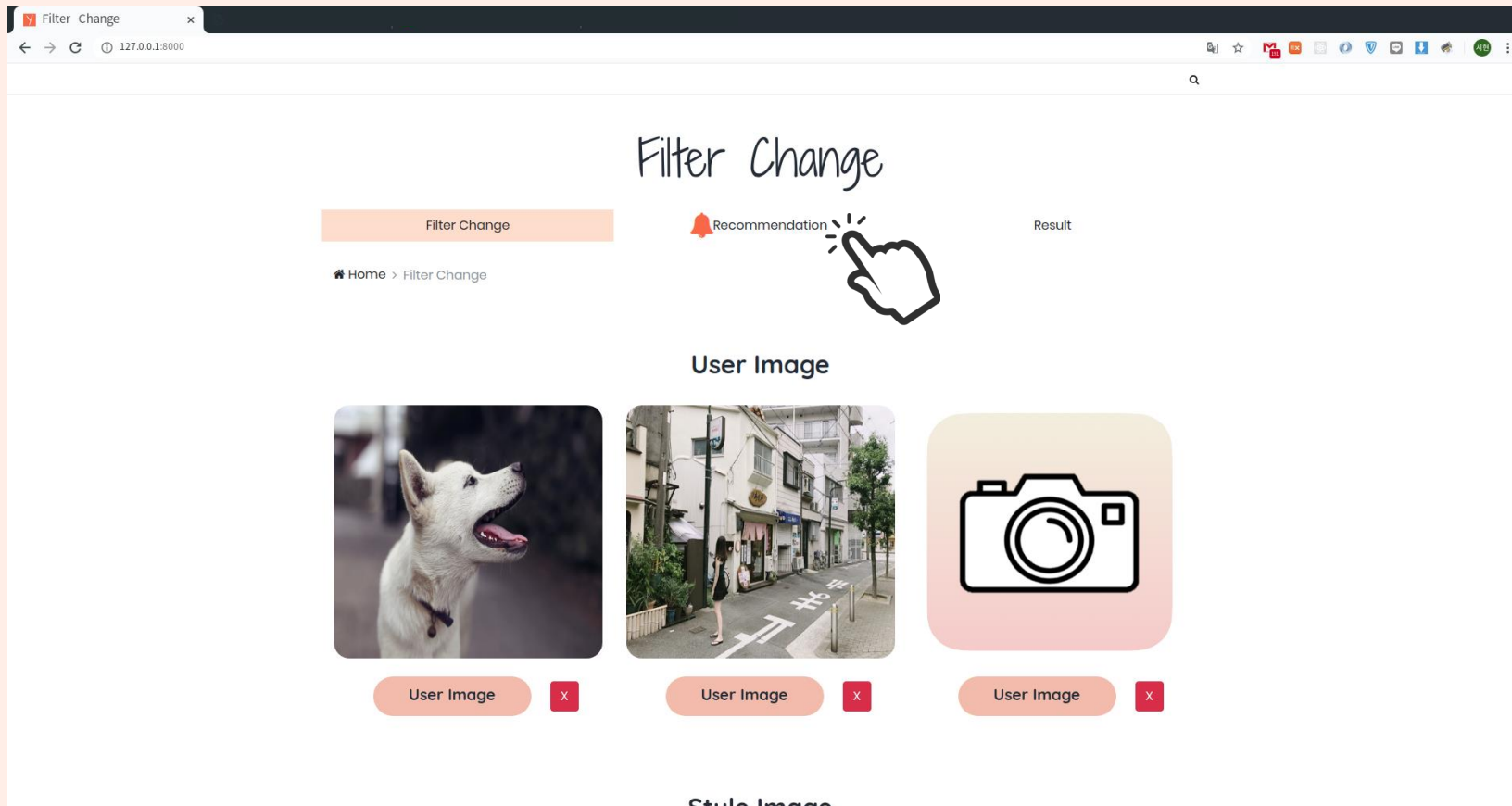


결과 이미지를 다운로드 하고
해당 파일을 열었을 때
보이는 화면이다.

원본 사진의 크기와 비율이
줄어들지 않고 보존되어 있는
것을 확인할 수 있고,
유저 이미지(User Image)에
스타일 이미지(Style Image)
의 색감이 입혀진 것을
확인할 수 있다.

2. 웹 서비스 사용

추천 이미지 확인

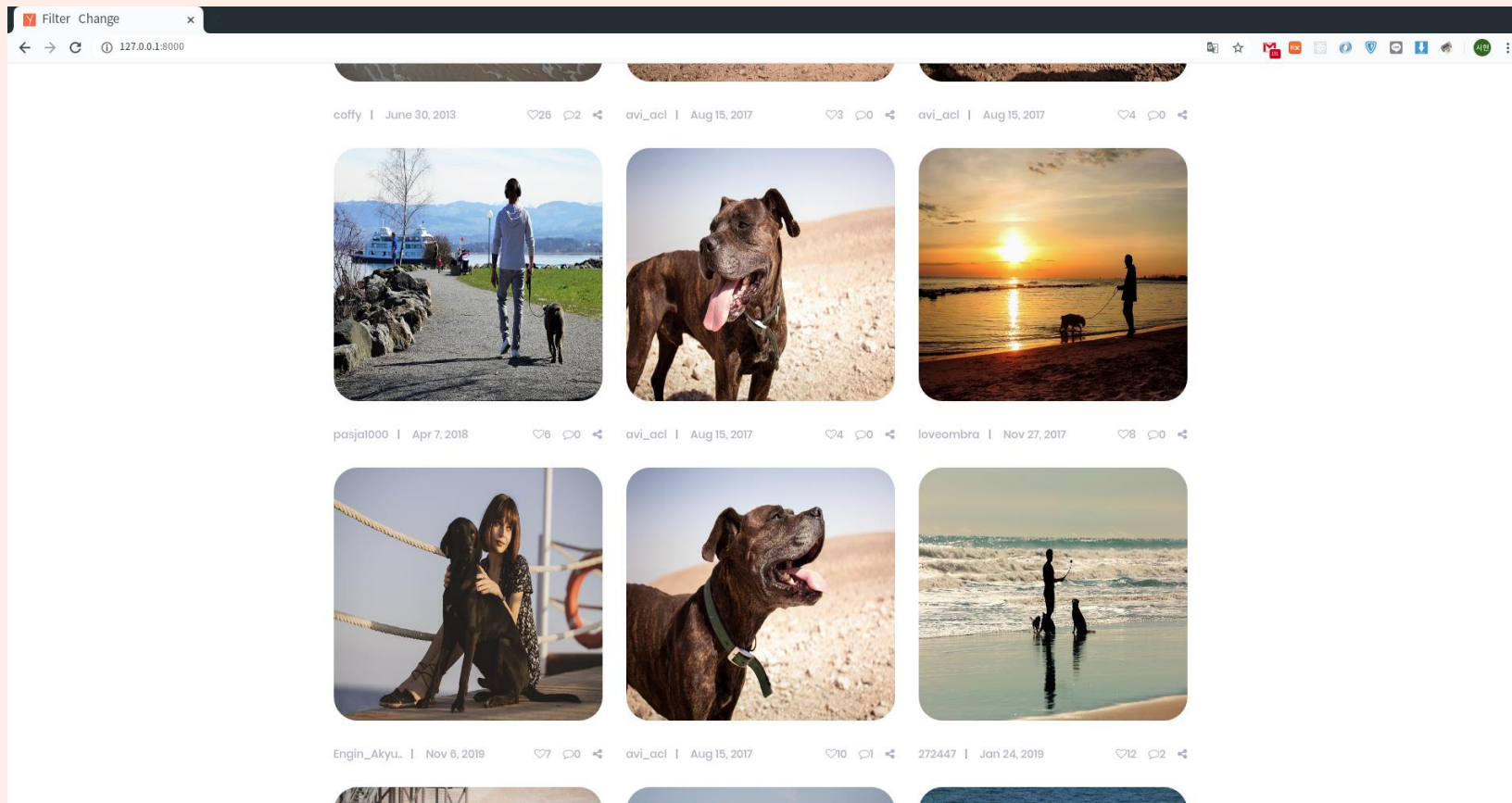


유저 이미지(User Image)를 등록하면 이를 기반으로 이미지의 키워드를 추출하여 추천 이미지들을 보여준다.

유저 이미지를 등록하면 Recommendation 탭에 불이 켜지는 것을 확인할 수 있다.

2. 웹 서비스 사용

추천 이미지 확인

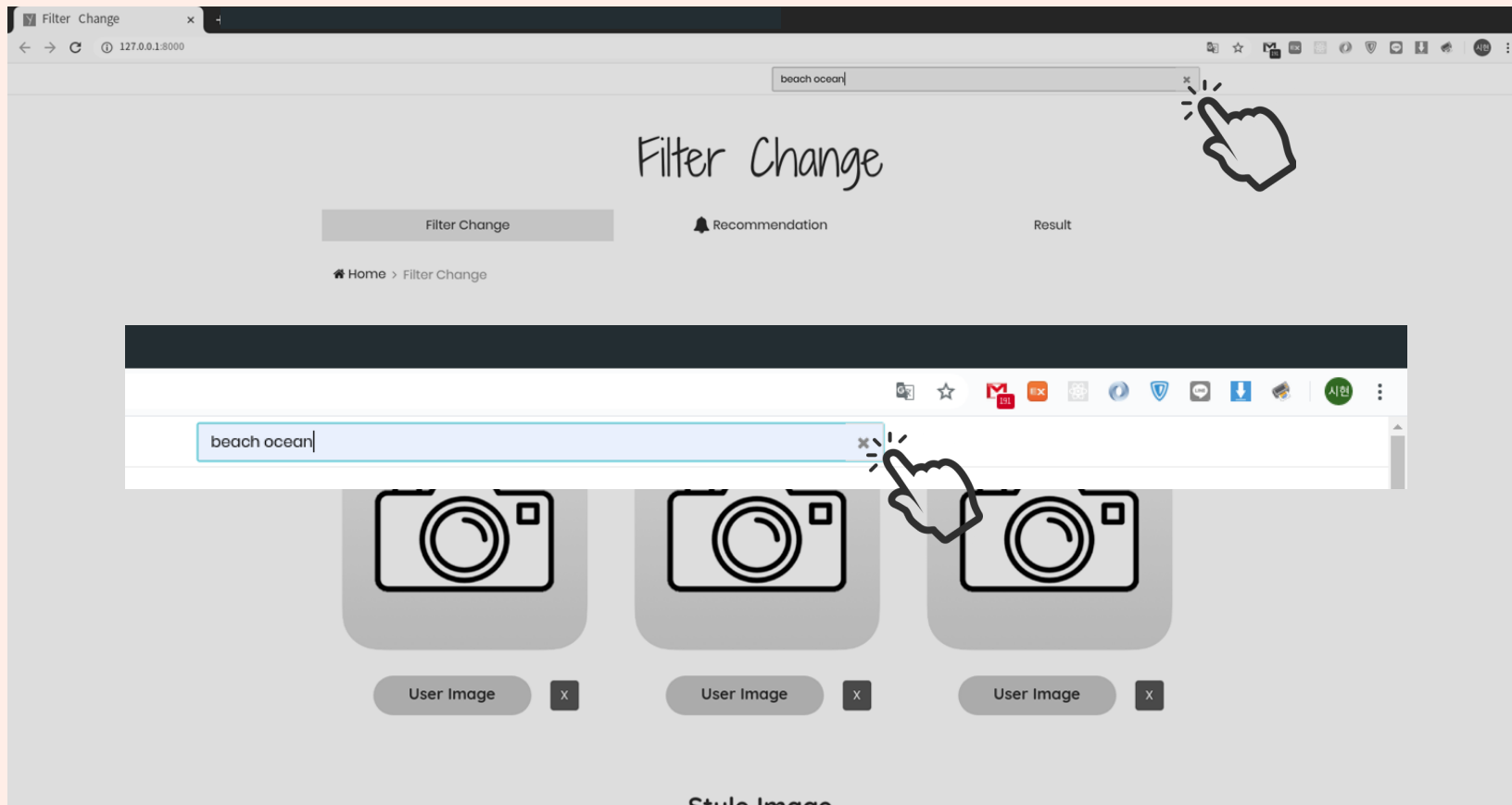


업로드 한
유저 이미지(User Image)
와 관련된 키워드들을
포함한 사진들이 추천된 것을
확인 할 수 있다.

또한, 각 이미지들의 정보를
확인할 수 있다.

2. 웹 서비스 사용

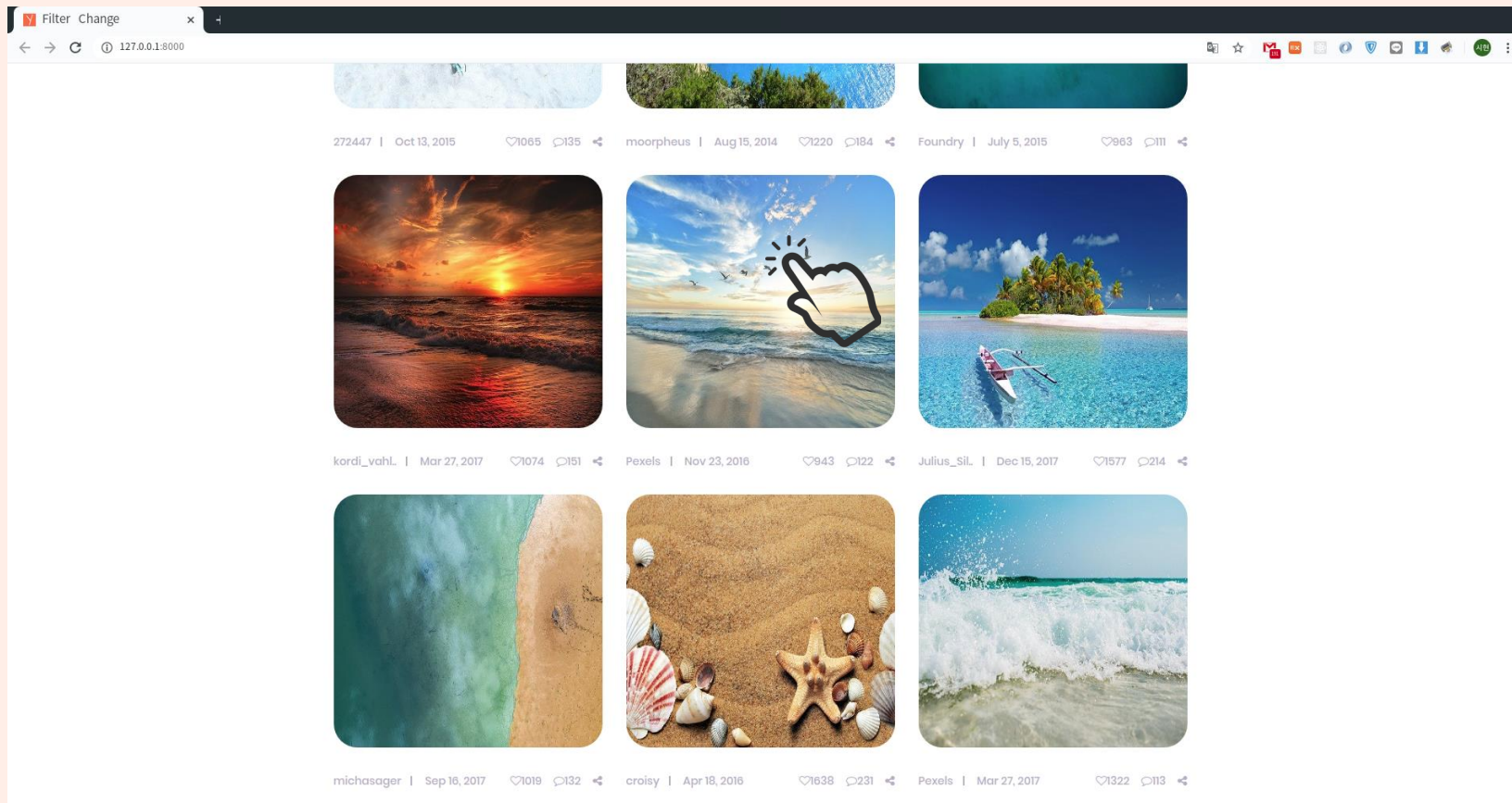
키워드 검색



페이지의 우측 상단에
키워드 검색을 할 수 있는
검색 창에서 찾고자 하는
이미지의 키워드를 입력하고
엔터(Enter)키를 누른다.

2. 웹 서비스 사용

검색 결과 확인

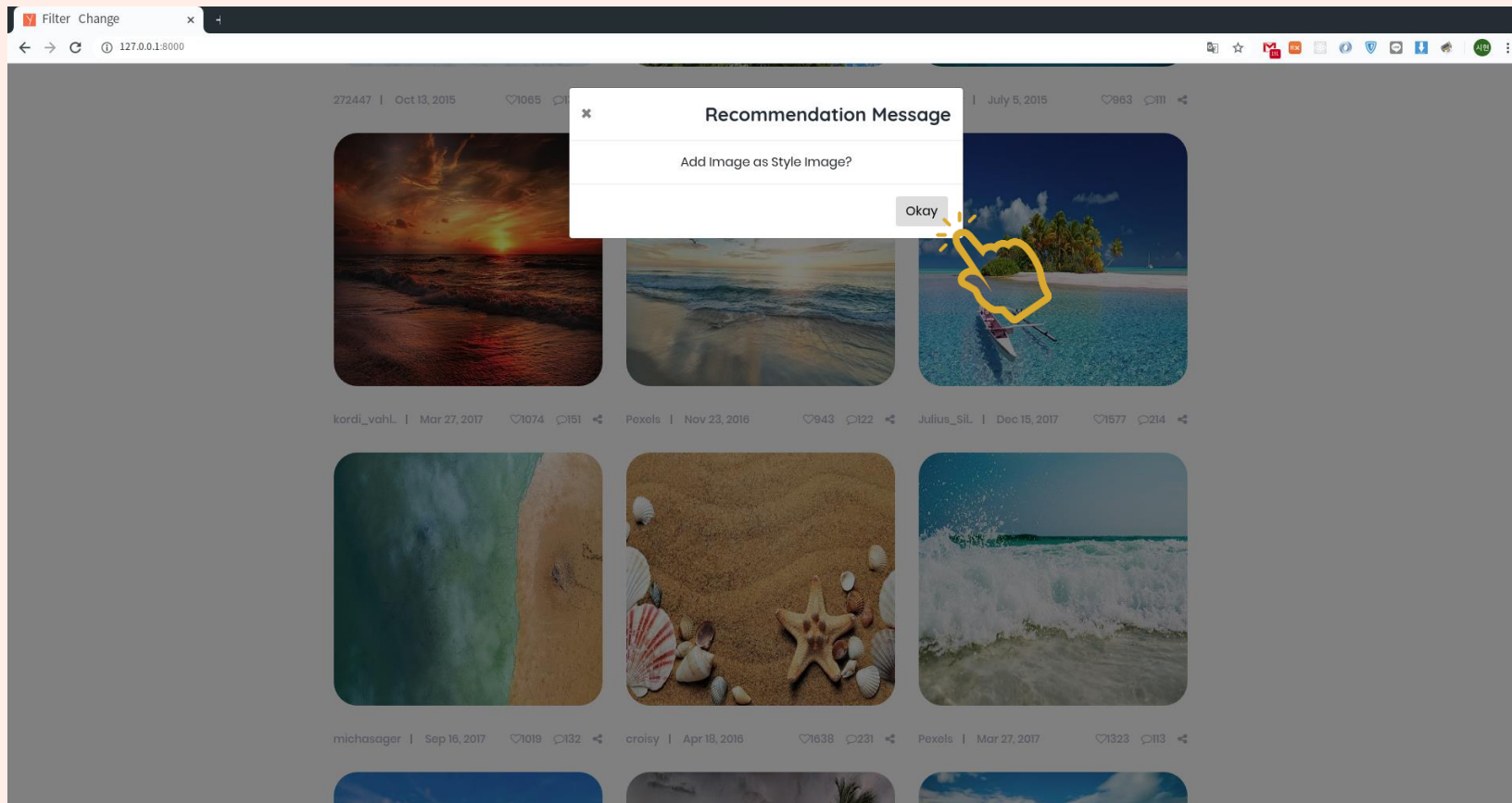


검색한 키워드를 포함한
이미지들이 검색된 것을
확인할 수 있다.

이미지 중 마음에 드는 사진은
스타일 이미지(Style Image)
로 적용시킬 수 있다.

2. 웹 서비스 사용

팝업 모달 확인



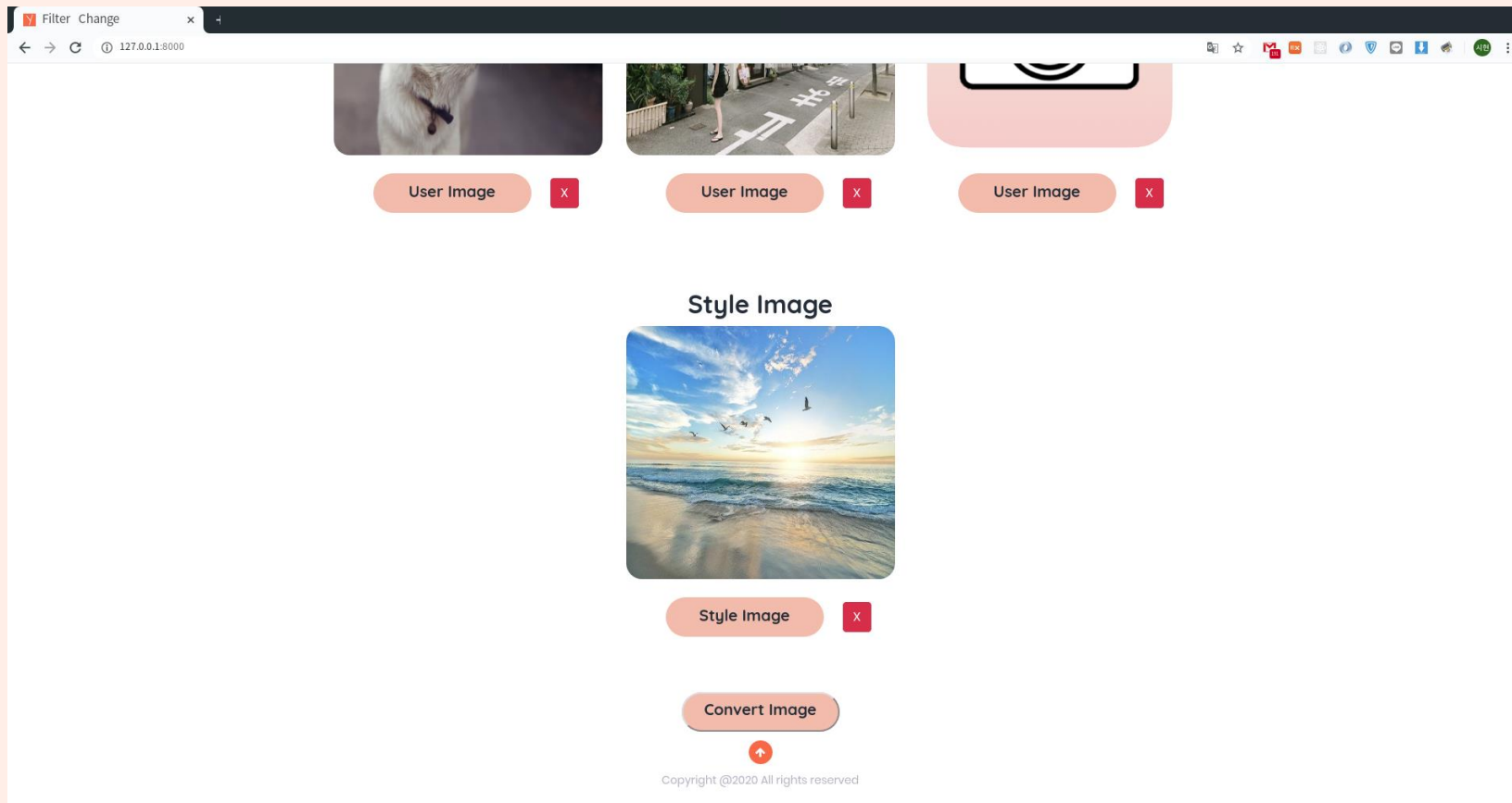
“해당 이미지를 스타일 이미지로 적용시키시겠습니까?” 라는 안내 문구를 확인한다.

스타일 이미지로 적용시키기 위해, Okay 버튼을 클릭한다.

되돌아가고 싶을 경우, 좌측 상단의 x버튼을 클릭한다.

2. 웹 서비스 사용

스타일 이미지 적용



적용된 스타일 이미지
(Style Image)를 확인하고,
유저 이미지(User Image)를
추가적으로 업로드 한 후,

Convert Image 버튼을 누르며
웹 서비스를 사용한다.